

N 9 4 - 1 8 3 6 1
7.1.1

3rd NASA Symposium on VLSI Design 1991

Systolic Array IC for Genetic Computation

D. Anderson
Hewlett Packard
Northwest IC Division
1020 NE Circle Blvd.
Corvallis, OR 97330
daryl@hpcvcan.cv.hp.com

1 Introduction

Measuring similarities between large sequences of genetic information is a formidable task requiring enormous amounts of computer time. Geneticists claim that nearly two months of CRAY-2 time are required to run a single comparison of the known database against the new bases that will be found this year, and more than a CRAY-2 year for next year's genetic discoveries, and so on.

The DNA IC, designed at HP-ICBD in cooperation with the California Institute of Technology and the Jet Propulsion Laboratory, is being implemented in order to move the task of genetic comparison onto workstations and personal computers, while vastly improving performance.

The chip is a systolic (pumped) array comprised of 16 *processors*, control logic, and global RAM, totaling 400,000 FETS. At 12 MHz, each chip performs 2.7 billion 16 bit operations per second. Using 35 of these chips in series on one PC board (performing nearly 100 billion operations per second), a sequence of 560 bases can be compared against the eventual total genome of 3 billion bases, in minutes — on a personal computer.

While the designed purpose of the DNA chip is for genetic research, other disciplines requiring similarity measurements between strings of 7 bit encoded data could make use of this chip as well. Cryptography and speech recognition are two examples.

A mix of full custom design and standard cells, in CMOS34, were used to achieve these goals. Innovative test methods were developed to enhance controllability and observability in the array. This paper describes these techniques as well as the chip's functionality.

This chip was designed in the 1989-90 timeframe.

2 Goals

The main project goal was to produce a device, for a larger system, that would prove the new computing architecture. This meant integrating as much functionality as was reasonable, with respect to cost. This includes as many processors, as much RAM per processor, and as many other desired functions as possible. Performance was a lesser concern, largely due to disk access being the initial system performance limiter, and also because the architecture provides the main performance breakthrough. Limiting power dissipation was a lesser, but real concern as well.

At the outset of the project, a standard cell implementation was envisioned that might contain 10 processors, each with 32 bytes of RAM on a 1 cm square device. In the end, a custom solution provided 16 processors, each with double the functionality and 128 bytes of RAM, on a roughly 1 cm x 1.2 cm device.

3 Functionality

The primary function of the system, comprised largely from a series of DNA chips, is to locate regions of similarity between strings of genetic bases, represented in ASCII by the characters; A, C, G, and T. A terse description of the method for achieving this end, is as follows.

First, the *primary string* is converted by an external processor, such that each character in that string is replaced by four *match scores*, one for each of the four possible characters that it may be compared against in the *secondary string*. These scores, for each character in the primary string, are loaded into the local RAMs of each successive processor, such that each RAM contains the four bytes representing the four possible scores caused by interaction of characters in the secondary string with that single character of the primary string. Each processor's RAM is 128 bytes, enough to accommodate full ASCII. Now, each processor behaves as the agent of one character in the primary string; hence, the length of the primary string is initially limited to the number of processors in the system (16 x number of DNA chips). Through software the length can be expanded without limit, by method of partitioning the string and using sufficient overlap.

Secondly, a number of constants are loaded into each chip by the external system processor, such as; chip location within the pipeline, how to deal with gaps that naturally occur within genetic sequences, and others.

At this point, the pipeline begins to function. The secondary string enters the front of the pipeline and is passed from one processor to the next on each successive clock. Each character within this string is used as an address to the local RAM of the current processor visited by that individual character. By this method, the appropriate score is retrieved from the local RAM for the interaction between the characters of the two separate strings. Along with the former occurring, the follow three equations are processed within that same clock cycle, in each processor. (Smith and Waterman, *Best Subsequence Alignments Algorithm*.):

$$H_{i,j} = \max\{0, H_{i-1,j-1} + s(a_i, b_j), E_{i,j}, F_{i,j}\}$$

with

$$E_{i,j} = \max\{H_{i,j-1} - (u_E + v_E), E_{i,j-1} - v_E\}$$

$$F_{i,j} = \max\{H_{i-1,j} - (u_F + v_F), F_{i-1,j} - v_F\}$$

where F, H and b are pipelined, E is fed back within the processor, u_E, v_E, u_F , and v_F are constants dealing with sequence gaps, and $s(a_i, b_j)$ is the score produced by the intersection of the two characters from the different strings.

Additionally, each processor monitors its H value, which represents quantitatively the similarity between strings, and detects its peak. If this peak exceeds a programmed threshold, this value, as well as the location of this occurrence within the secondary string, is piped along through the remaining processors on the given chip and then stored in the chip's global FIFO RAM. The range of this location value limits the secondary string to 4 million characters. However with use of external software, an unlimited string can be applied.

This process occurs simultaneously in all processors on each chip, until the entire secondary string has been piped all the way through, or the external system processor interrupts. The equations and peak detector are implemented with five adders and seven comparators; the values $(u_E + v_E)$ and $(u_F + v_F)$ are provided as constants.

When a value has been stored in the global FIFO RAM, the chip signals the external system processor, and at the system processor's convenience, reads that data from the chip into a global system RAM. This is the raw similarity information desired from the system. Of course, if any chip's FIFO nears overflow, a system interrupt is issued, by that chip, to pause the entire pipeline until the RAMs can be emptied.

4 Design Challenges

Technical challenges included; performance, power, and density concerns, as well as problems pertaining to pad switching noise and testability.

By custom designing most circuitry for near maximum density, lower power and higher performance fell out as by-products. Most N channel devices in the pipeline were sized at 5μ wide and 1μ long. The small devices reduced power consumption, as well as greatly improved the circuit density. By careful floorplanning to minimize interconnect capacitance, chip performance was improved over that of a standard cell approach. One of the key sub-modules within the processor is a 16 bit adder. At 426μ by 215μ (896 FETs), the custom adder is one seventh the size of its standard cell implementation; at 4 mW, its power consumption is one sixth; and at 11 nS, its performance is improved by more than two fold over the standard cell solution.

While the conservative design goal of 12 MHz does not seem worthy of CMOS34, consider two of the paths to be traversed in the 83 nS cycle; 1) Register — 16 bit signed addition — 6 x (16 bit signed compare and select greatest) — 5 gates — Register, and 2) Register — address RAM — 16 signed bit addition — 3 x (16 bit signed compare and select greatest) — 5 gates — Register.

The next area of concern was with pad switching noise. This resulted from being bound to a 208 lead Quad Flat Pack, with 190 signal pins, leaving only 18 power pads. While having a full synchronous design helped in some aspects, it also created the possibility of having all 65 pipeline outputs and all 32 global data bus pads switch simultaneously. Additionally, several other system pads could be switching as well. It is helpful that all pipeline input signals are latched on the rising clk, while the pipeline outputs do not change until a number of gate delays later. However, several volts of supply noise could easily be

7.1.4

generated by switching the standard pads, causing erroneous inputs and outputs on the global system pads. Additionally, latchup was a concern.

The solution was to create three modified pads and use an expanded power distribution scheme. All pad input sensors (TTL level Schmitt) were connected to one Vdd pad and two Gnd pads; 124 inputs total. The output drivers for pipeline outputs, and global data bus were connected to 4 Vdd pads and 5 Gnd pads; 97 outputs total. The remaining 4 global system pads, capable of causing system interrupts, were connected to their own isolated pair of power pads. Lastly, the chip core and output pad stage-ups were placed on two pairs of power pads.

While this helped to isolate the noisy circuitry from sensitive circuitry, the noise spikes on the *dirty* power bus from output switching, were still too high. Several things were done to help reduce the noise. First, the drivers for the 65 pipeline outputs were greatly reduced so that the rise time on the Sentry 15's 60 pF load would be 40 nS worst case. These outputs will normally see only 7-10 pF in the product, as the output pad communicates only with the neighboring chip's input pad.

The global data bus pads created another problem in that their loading depended directly on how many DNA chips were placed in the system, as they all connect directly to one another. In the initial system, this load would be 275 pF. Since the 32 data bus pads were by far the largest contributor to noise, and because their load could vary, another scheme was employed. The data pads each contain two sets of output drivers; one small and one large. A signal to the pad determines whether the large drivers are used in parallel with the small ones, or whether the small drives are used alone. A control register bit is used to turn off the larger drivers, in the event that the data bus had a small capacitive loading, or that the noise from the larger drivers was simply unacceptable (in which case, the system clk rate would have to be reduced). The rise time for a 275 pF load with the large drivers is 20 nS, worst case, and 75 nS without those drivers.

Additionally, care was taken to turn on output drivers slowly; about a 3 nS to 4 nS rise time on the driver's gate. Skewing of data to the pad drivers also helped to reduce the switching noise.

The last major challenge was in the area of test. Standard methods for testing the part in its normal operating mode were seen to be near impossible. The controllability and observability of nodes deep in the pipeline of 16 processors was very near zero. Since each processor interfaces to the previous processor through a register bank, scan testing seemed to be the obvious solution. However, with about 150 register bits in each processor, and a total of 16 processors and one additional pipeline register buffer, the full scan vector length would be over 2500 bits. With a Sentry 15 limited to 256k total vectors, this would provide only 100 scan vectors, with no vector memory available for testing the 22k bytes of RAM, nor the control logic. Several thousand scan vectors were desired for testing the processors.

The solution was to take advantage of the fact that all of the processors are identical, and therefore given the same input scan vector, will produce the exact same resultant output vector in the register bank of the pipeline's next stage. The method then, is to scan in a vector that is only one processor register bank long (150 bits), into all 16

processors simultaneously. After clocking the device once in normal mode, as in standard scan methodology, the 16 resultant vectors are scanned out of the processors and onto 16 independent lines of the global data bus, readable from the chip's data bus pads. Additionally for testing in the product, all 16 scan outputs are connected, on chip, to a equality function. If, when in test mode, all of the 16 scan outputs are not equal, then an error pin is activated for notification of the external system processor. The system processor can then set another pin on the errant chip so that the pipeline data coming on chip is diverted around the 16 processors, to the final buffer register, thereby fixing the whole pipeline at the cost of those 16 processors.

5 Results

First prototypes of the **DNA** chip were tested in Spring of 1990. Several timing problems were found in chip functions that had not been completely simulated by the designers. Second prototypes produced perfect parts. JPL currently has a circuit board 16 **DNA** chips (a total of 256 processors) running and interfaced to a workstation.

6 Acknowledgments

- Ed Chin and Mike Yoo of JPL for their design work on the **DNA** chip.
Tim Brown, Paul Liebert, and Bobbie Manne of HP-ICBD Design Center Services for helping with artwork generation.
- Joe Casprowiak, Joan Long, Jimmy Packer of the HP-ICBD layout group.
- Fred Perner and Lynn Roylance of HP Labs for initial work on the **DNA** investigation.

